基于有效上下文信息的变体词还原方法 *

游绩榕[1,2]，沙　瀛[1,2]，梁　棋[1,2]，王　斌[1,2]

(1. 中国科学院信息工程研究所 第二研究室，北京 100093; 2. 中国科学院大学 网络空间安全学院，北京 100049)

摘　要：在社交网络上，用户常创造一些变体词来替代部分实体名词，将这些变体词还原为原目标词是自然语言处理中的一项重要工作。针对现有变体词还原方法准确率不够高的问题，提出了基于有效上下文信息的变体词还原方法。该方法利用点互信息抽取出变体词和候选目标词的有效上下文信息，并将其融合进自编码器模型中，获得变体词和候选目标词更准确的编码，并依据此计算相似度进行候选目标词排序，更准确的实现了变体词还原任务。实验表明，该方法较当前主流的几种方法相比效果有显著提升，提高了变体词还原的准确率。

关键词：变体词；变体词还原；自编码器；有效上下文信息；词嵌入；神经网络

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.01.0033

Morph resolution based on effective context information

You Jirong[1,2], Sha Ying[1,2], Liang Qi[1,2], Wang Bin[1,2]

(1. *Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100093, *China*; 2. *School of Cyber Security*, *University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)

Abstract: In social networks, people often creates morphs to replace some entity names. How to resolve these morphs to their real target entities is a very important task for natural language processing. In order to overcome the shortcomings that existing methods cannot resolve morphs accurately, this paper proposed a morph resolution method based on effective context information. This method extracted the effective context information of morphs and target candidates, and integrated the effective context information into autoencoders in order to get more accurate embedding of morphs and their target candidates. This method then calculate the similarity between morphs and target candidates based on the accurate embeddings, and ranked the target candidates according to the similarity. The experiments show that this approach significant outperforms the state-of-the-art methods and improves the accuracy of morph resolution.

Key words: morph; morph resolution; autoencoder; effective context information; word embedding; neural network

0　引言

变体词在互联网中广泛存在。在互联网上，人们常常把一些规范的专有名词，如人名地名，通过各种方式改造，创造一些不规范的词汇来替代原来的词，来规避审查、或表达讽刺、娱乐等情感，这就是变体词现象。这些创造出来的新词就叫做变体词，与之对应的是原来的词，即目标词。例如图 1 中的这条微博：“勇士创造了属于自己的时代潮流，大势所趋连规则也跟着改变，其他球队也跟着效仿。詹皇自带的体系，跟潮流不同，他也不愿苟同。小球时代其实是算术题，没了血性，没了对抗，没意思了。”这条微博中，“詹皇”就是一个变体词，它指代的是球员“詹姆斯 ”“詹姆斯”就是它的原目标词。

> 勇士创造了属于自己的时代潮流，大势所趋连规则也跟着改变，其他球队也跟着效仿。詹皇自带的体系，跟潮流不同，他也不愿苟同。小球时代其实是算术题，没了血性，没了对抗，没意思了。

图 1　变体词在微博中的使用

对变体词的研究在自然语言处理中具有实际的意义。自然语言处理的基础就是正确的对词语的分析和理解，传统分析方法有赖于人工整理的词典、词林等资源，但是在面对语言灵活、变化快速的社交媒体语言时，传统方法会遇到很多困难。互联网创造新词汇的速度很快，变体词就是其中一类，它们通常不会出现在词典中，也缺乏释义和理解，在词法分析时会产生干扰。如果能够将变体词都还原成它们的目标词，能够增加词法分析的准确性，为下游其他的自然语言处理任务提供支持。

收稿日期：2018-01-18；**修回日期：**2018-03-01　　**基金项目：**科技部“十一五”科技支撑计划资助项目（2017YFB0803301）

作者简介：游绩榕（1993-），男，福建福州人，硕士研究生，主要研究方向为自然语言处理，文本分析；沙瀛（1973-），男（通信作者），副研究员，博士，主要研究方向为社会计算、复杂网络（youjirong@iie.ac.cn）；梁棋（1989-），女，助理研究员，博士研究生，主要研究方向为信息检索、舆情计算；王斌（1972-），男，研究员，博士，主要研究方向为信息检索、自然语言处理、数据挖掘.

为了还原变体词，目前有几类主流的解决的思路。一类是基于建立规则实现，包括语音替换、汉字拆分等等[1][2][3]。这类方法的优点是简单直接，但是变体词变化方式繁多且不断变化，规则的适用性很有限。另一类是基于统计和规则的方法，将统计的方法与规则相结合，通过提取一些特征使用统计学习的方法进行变体词还原[4~6]。这类方法相比直接建立规则要更加灵活，但是统计学习的方法是重度依赖特征的，仍需要大量的特征工程。此外，还有一种是基于语义表示的方法。语义表示的方法是建立在分布假说[7]上的，利用分布假说可以通过上下文对变体词进行建模，从而实现变体词的还原。语义表示的方法需要建立较为复杂的模型，但能达到比较好的效果。

本文沿用了语义表示的思路来解决变体词还原问题。目前基于语义表示进行变体词还原的研究都只是简单的利用临近的上下文词项，然后直接套用通用的词向量模型，缺乏对上下文信息进行有效的筛选。本文提出一种基于有效上下文信息的变体词还原方法，通过计算上下文与词项间的互信息来筛选出有效上下文信息，然后使用自编码器模型将词项和它的有效上下文信息进行融合，得到联合编码。得到的编码即可用于计算变体词与其他词的相似度，根据相似度排序即可实现变体词还原的任务。本文对此方法进行了实验验证，对比了当前效果最好的几种变体词还原方法。实验结果表明本文的方法是有效的，相比当前最好的方法精确率得到有效提升。

本文的主要贡献有：a）提出了基于有效上下文信息的变体词还原方法，有效提升了变体词还原的准确性；b）利用词语和上下文词项间的互信息来筛选有效上下文信息；c）利用联合上下文的自编码器对词语及其有效上下文信息进行联合编码，使之更好的表示词语之间相似性。

1　相关工作

关于变体词的研究最早出现在一些关于不规范文本或网络语言的规范化的工作中。例如 Wong[1]的工作中研究了中文网络聊天中由语音变化产生的字词替换现象，例如将“我”替换为“偶”，这与变体词现象很类似。早期的不规范文本规范化主要使用基于规则的方法，例如 Wong[1]、Xia[2]、Sood[3]等人的工作。后来一些研究提出可以结合统计和规则进行还原，如文献[4~6]的工作。典型的方法如 Wang[4]的工作，Wang 基于拼音、缩写、替换等典型特征，建立一个概率模型，用概率模型进行监督训练实现不规范文本的还原。

在 Huang 等人[8]的研究中，变体词的概念首次明确出现。Huang 等人[8]研究了变体词的基本特征，包括表面特征、语义特征和社交特征，根据这些特征设计了简单的分类模型进行变体词的还原，并验证了效果。在这之后，Zhang 等人[9]提出了一个端到端的变体词解码方案，它同时进行变体词发现和变体词还原两个任务，先按照一定标准在大量语料中找到其中的变体词，然后再将它们还原为目标词。还原的过程利用了神经语言模型和词嵌入方法，将文本中的词项，包括变体词和目标词，都编码为词向量，然后比较计算它们的相似度，再按照候选词与变体词的相似度进行排序来得到变体词还原的结果。

除此之外，还有一些关于变体词自动生成的研究。Hiruncharoenvate 等人[10]通过新浪微博的语料研究了用户如何通过使用同音异形异义的变体词来规避审查，并尝试使用非确定性算法生成大量新的变体词。Zhang 等人[11]更进一步的研究了变体词的特征，并尝试自动生成变体词的样本，从另一个角度来解决变体词还原的问题。研究总结了八大类变体词产生的模式，包括拼音、拆字、昵称、翻译等等模式。Sha 等人[12]提出了基于字词联合的变体词还原方法，在将词语进行编码的同时还对字进行了编码，联合两种编码解决变体词还原问题。

本文使用了联合上下文的自编码器。自编码器是一种无监督的神经网络，它能够对输入向量进行编码，然后进行解码重建，从而获取输入向量有用的特征融入编码之中。自编码器有许多变种。联合上下文的自编码器[13]将词语的上下文连同词语一起输入自编码器中，得到它们的联合编码。

2　基于有效上下文信息的变体词还原方法

在阐述本文的变体词还原方法前，本文先形式化定义变体词还原问题：变体词还原指的是给定一组变体词，找到每个变体词可能的候选目标词列表，并根据可能性进行排序。给定文档集合 $D=\{d_1,d_2,\ldots,d_{|D|}\}$ 与变体词集合 $M=\{m_1,m_2,\ldots,m_{|M|}\}$ 。在 D 中找出每个 m_i 可能的候选词，并进行排序，得到 m_i 对应的候选目标词列表 $\hat{T}_{mi}=\{t_1,t_2,\ldots,t_{|T|}\}$，其中排名越前的候选目标词越可能是 m_i 真正的目标词，即完成了变体词的还原。

以图 1 中的例子为例，变体词为“詹皇”，本文首先需要在微博中找到这个变体词的候选目标词，然后进行排序得到目标词列表。而“詹皇”真正的目标词“詹姆斯”应该在目标词列表中越靠前越好。

本文提出的基于有效上下文信息的变体词还原方法是一种基于语义表示的算法。本文将变体词和它的候选目标词进行编码，利用编码进行排序。与其他方法所不同的是，本文的算法筛选出了有效上下文信息，而不是简单的使用临近的上下文词项。这能够帮助本文更好的找到与变体词含义相同的目标词。

整个算法大致分为四个过程。如图 2 所示，首先，从文档集合中先进行初步筛选，找到变体词的候选目标词列表；其次，抽取出这些词语的有效上下文信息，包括变体词以及它们的候选目标词；然后，使用联合有效上下文信息的自编码器将各个词语和它们的有效上下文信息进行联合编码，得到变体词和各个候选目标词的编码表示；最后，计算变体词的编码和候选目标词的编码的相似度，并按照相似度对候选目标词进行排序，从而完成变体词还原的任务。

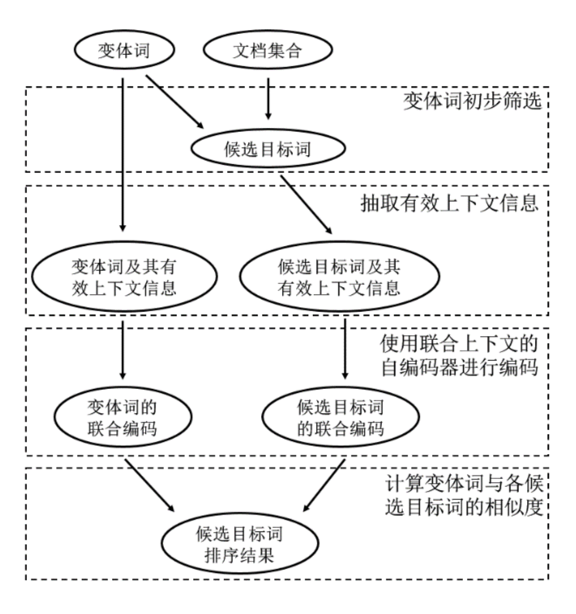


图 2 基于有效上下文信息的变体词还原方法的流程

2.1 候选目标词的初步筛选

变体词还原的第一步是候选目标词的初步筛选。本文主要利用两条标准来实现。

a)利用时间共现性筛选。统计表明，变体词与其目标词通常都会同时在一个较短的时间段内多次出现[12]。因此本文可以分析变体词出现的时间，然后只需要在附近的时间段里的词语中寻找目标词，即可减小候选词集合大小。具体过程是：对于给定的变体词，本文可以找到含有变体词的文档（例如微博），根据这些文档的发布时间可以设定一个时间窗口，本文寻找这个时间窗口内的文档，在这些文档中寻找候选目标词。

b)根据词性筛选。由于变体词所指代的往往都是人名、地名和组织名等专有名词，所以候选目标词也只需要在专有名词中寻找。专有名词的筛选可以通过词性标注和命名实体识别即可得到，有许多成熟的工具，包括 NLPIR[14]、Stanford NER[15] 等，均可完成这个任务。联合时间共现性的筛选结果即可得到候选目标词。

2.2 抽取有效上下文信息

变体词还原的第二步是抽取变体词和各候选目标词的有效上下文信息。上下文信息在各种基于词嵌入的方法中被广泛使用，但是这些方法中只是简单的使用某词语临近的上下文词项并套用词嵌入模型，而没有对上下文词项进行筛选。实际上，并非某个词语所有的上下文词项都与它有密切的语义联系。而抽取出与词语有较强语义联系的上下文词项可以帮助本文更好的进行变体词与候选词之间的相似度判断。本文称这种上下文词项构成的集合有词语的有效上下文信息。

本方法中判断有效上下文信息的标准是词语与上下文词项间的点互信息（PMI）。点互信息描述了词语之间的共现关系，如果点互信息越高，那么两个词共现频率越高，语义联系越大；反之则语义联系越小。点互信息的计算公式为

$$PMI(x;y)=log\frac{p(x,y)}{p(x)p(y)} \qquad (1)$$

其中: x 和 y 是指代两个词语， $p(x)$ 和 $p(y)$ 分别表示 x 和 y 在语料中出现的概率， $p(x,y)$ 为 x 和 y 在语料中共现的概率。

利用点互信息可以很好的判断词语与其上下文词项的语义联系。举个例子，有“韦德”“闪电侠”和“贝克汉姆”三个词项，其中，“韦德[1]”是变体词“闪电侠”的目标词，而“贝克汉姆[2]”与“闪电侠”无关，不是它的目标词。本文挑选出三个它们的上下文词项：“波什”、“詹姆斯”、“体育”进行比较。其中“波什”、“詹姆斯”均与“闪电侠”、“韦德”有较强语义联系，与“贝克汉姆”语义联系较低。表 1 展示了不同的上下文词项的 PMI 值，可以看出，“波什”、“詹姆斯”与“闪电侠”、“韦德”的点互信息都比较高，与“贝克汉姆”的点互信息较低。点互信息可以很好的表示词项之间的语义联系。

此外可以发现，“波什”、“詹姆斯”与“闪电侠”、韦德”的点互信息较高，与“贝克汉姆”的点互信息较低，有较强的区分性；而词项“体育”与这三个词的点互信息相差不大，区分性较低。因此本文可以看出通过点互信息筛选出的有效上下文词项相对于其他的词项，能够更好的区分出意义不同的词语，找到意义相同的词语。这个性质能够很好的帮助本文找到变体词真正的目标词。

表 1 有效上下文词项与其他词项的对比

PMI	有效上下文词项 波什	 詹姆斯	其他词项 体育
闪电侠	**3.12**	**2.46**	0.12
韦德	**9.79**	**9.27**	0.37
贝克汉姆	0.34	0.63	0.49

本文使用点互信息上下文过滤器对词项的上下文进行过滤来生成词项的有效上下文信息。对于词项 w，先在全局范围内取窗口 wd 内的词项。注意到先去掉助词、介词等类型的词项，这些词项显然不会是有效上下文词项。得到的上下文词项形成集合 $C=\{c_1,c_2,\ldots,c_{|C|}\}$，计算每个词项 c_i 与 w 的点互信息 $PMI(w,c_i)$；然后，取集合中最大的前 K 个，作为有效上下文词项集合 FC_w，从而得到有效上下文信息。

2.3 对变体词和候选目标词进行编码

在抽取有效上下文信息之后，需要融合词语和它的有效上下文信息特征，得到一个联合编码。使用联合有效上下文信息的自编码器来进行编码。图 3 展示了自编码器的结构和编码流程。联合有效上下文信息的自编码器也是由多个基本自编码器构成。图 3(a)为基本自编码器结构。自编码器的输入 x 经过编码后得到编码表示 h，然后再进行解码得到输入 x 的精确重建 $\hat{x}$，即

$$h=g(Wx+b) \qquad (2)$$

$$\hat{x}=g(W'h+b') \qquad (3)$$

其中式（2）是自编码器的编码过程，式（3）是自编码器的解码过程 $W\in R^{d\times d'}$ 和 $b\in R^{d'}$ 为编码器的学习参数，$W'\in R^{d'\times d}$ 和 $b'\in R^{d}$ 为解码器的学习参数， g 为激活函数。 d 和 d' 分别表示输入的维度和编码之后的维度，通常 $d'<d$，以达到压缩和降维的目的。自编码器的优化目标是使得 x 和 $\hat{x}$ 的差异尽量的小，这样编码

[1] 韦德，美国篮球运动员

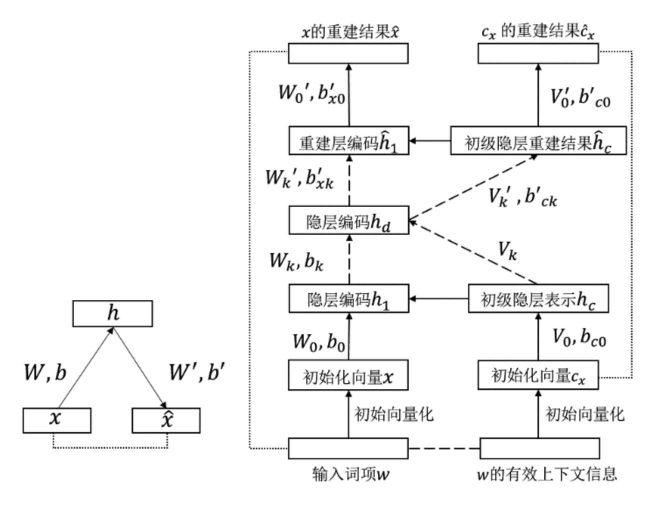
[2] 贝克汉姆，英国足球运动员

表示 h 就可以准确有效表示 x 。通常会使用平方损失函数作为优化目标，即

$$l(x)=\|x-\hat{x}\|^2 \qquad (4)$$

优化目标为

$$\min_{\Theta}\sum_{i=1}^{n} l(x^{(i)}),\ \Theta=\{W,W',b,b'\} \qquad (5)$$



（a）基本自编码器　　（b）联合有效上下文信息的自编码器

图 3 联合有效上下文信息的自编码器结构图

基本的自编码器只能输入单个向量 x 。因此本文对基本自编码器进行了拓展，并使用了多层的层叠式自编码器结构，来融合输入词语与其有效上下文信息。如图 3(b)，首先，本文使用词嵌入的方法进行初始向量化，得到输入词项 w 与其有效上下文信息初始向量。 w 的初始向量为 x ，而对于有效上下文信息，对其中有效上下文词项分别进行初始向量化，然后取这些向量的平均值作为有效上下文信息的初始向量，记作 c_x 。使用经典的词嵌入方法进行初始向量化，例如 Word2Vec[16]、GloVe[17]等，这也是很普遍的一种做法。初始向量化之后，用 c_x 学习出上下文向量的初级隐层表示 h_c ，这一步可由基本自编码器来生成；然后将 h_c 和 c_x 输入到拓展的层叠式多层自编码器中进行编码和重建。编码阶段，第 k 层的自编码器单元的输入为上一个自编码器单元的编码结果 h_{k-1} （如果 k=1，则输入为 x）和上下文向量的初级隐层 h_c ，输出为本层的隐层 h_k ，一个 d 维的编码表示，即

$$h_k=g(W_k h_{k-1}+V_k h_c+b_k) \qquad (6)$$

其中： W_k,V_k,b_k 表示将 h_{k-1} 和 h_c 编码成 h_k 的参数；解码阶段，每个隐层将其编码结果 h_k 分别重建，得到对应输出两个重建结果：上一层的隐层的编码 h_{k-1} 和上下文向量的初级隐层 h_c ，即

$$\hat{h}_{k-1}=g(W_k' h_k+b_{xk}') \qquad (7)$$

$$\hat{h}_c=g(V_k' h_k+b_{ck}') \qquad (8)$$

其中： W_k' 和 b_{xk}' 表示将 h_k 重建为 h_{k-1} 的参数， V_k' 和 b_{ck}' 表示将 h_k 重建为 h_c 的参数。最后由 h_1 和分别得到最后的重建结果 $\hat{x}$ 和 $\hat{c}_x$ 。

自编码器融合了输入词项和它的有效上下文信息，其优化目标需要让 $\hat{x}$ 和 $\hat{c}_x$ 的重建误差都比较小。因此设置自编码器的损失函数为

$$loss(x,c_x)=\|x-\hat{x}\|^2+\lambda\|c_x-\hat{c}_x\|^2 \qquad (9)$$

其中 $\lambda\in[0,1]$ 是调节上下文信息在编码中影响的权值。则优化目标为

$$\min_{\Theta}\sum_{i=1}^{n} loss(x^{(i)},c_x^{(i)}),$$
$$\Theta=\{W_k,W_{k'},V_k,V_{k'},b_k,b_{xk}',b_{ck}'\}, \qquad (10)$$
$$k\in 1,2,\dots,depth$$

根据优化目标，利用神经网络通用的训练方法如随即梯度下降法等，可以对联合有效上下文信息的自编码器进行训练。自编码器是个无监督的模型，因此只需要将大量语料中的词项作为样本依次输入模型中，即可完成训练。该自编码器的输出为编码阶段最后一个隐层的编码表示 h_{depth} 。 h_{depth} 是个联合编码，融合了输入词项和它的有效上下文信息的特征，可以更好的表示词项之间含义的相似度，帮助本文更准确的找到变体词真正的目标词。

2.4 对候选目标词进行排序

得到变体词和候选目标词的编码表示之后，就可以对候选目标词进行排序。自编码器将词语映射到一个向量空间中，用向量的余弦相似度大小来判断词语的相似性。相似度越大，词语越相似。对于每一个变体词 m_i ，计算它的每个候选目标词 t_i 与它的余弦相似度，然后按照相似度排序，即可得到 m_i 候选目标词的排序结果 $\hat{T}_{mi}=\{t_1,t_2,\dots,t_{|T|}\}$ ，从而完成了变体词还原的过程。

3 实验与分析

3.1 实验数据集

本文在 Huang 等人研究所用的数据集[8]的基础上进行了筛选，删去了一些对应关系错误的或出现次数太少的变体词，新增加了一些变体词，并采集了与这些变体词相关的微博，形成一份新的数据集。这份数据集中，共包含 1,597,416 条新浪微博消息，25,003 条 Twitter 消息。数据集中共含有 593 对变体词。

3.2 参数设置

关于模型中的参数选取，一部分沿用了比较经典的工作中的选择，另一部分通过验证集来选取效果最佳的参数。在候选目标词初步筛选时，参考 Sha 的工作[12]，取微博的时间窗口为 1 天，Twitter 的时间窗口为 3 天，获取各个变体词的候选词项集合。在使用自编码器进行编码时，本文使用 Word2Vec 方法来进行初始向量化，初始向量的维度为 100 维。

其余的各个参数通过验证集进行选取。随机选取了 50000 条微博作为验证集来调整参数。经过在验证集上测试，在抽取有效上下文信息时，取窗口 $wd=20$ ，向量数 $K=10$ 。在使用自编码器进行编码时，编码器的深度 $depth=3$ ， h_c 编码表示的维数 $d=100$ ，取 $\lambda=0.5$ 。

3.3 结果分析

由于得到的还原结果是一个排序，因此使用 $precise@k$ 这个指标来评价变体词还原的效果。本文中 $precise@k=N_k/Q$ ，对于每个变体词 m_i ，将它对应的目标词 e_{m_i} 在本文给出的排序序列出现的位置记作 p_i 。 N_k 在所有的变体词测试样本中， $p_i\le k$ 的变体词样本数量，Q 为所有变体词测试样本数量。若 $p=1$ 则说

明得到的候选目标词排序列表的第一位即是真正的目标词，即准确还原了这个变体词。

实验与目前效果最好的几种方法进行了比较，包括文献【8，9，12】的方法。本文中的方法记做 AE-ECI。图 4 和表 2 展示了几种方法在数据集上还原效果。从结果中可以看出，相比之前的方法，本文的方法在精确率上有一定的提升。对于 pre@1，本方法相比效果最好的 Zhang 的方法提升 3.41%，而对于 pre@10，本文的方法较最好的 Sha 的方法提升了 6.43%，显著提高了变体词还原效果。

表 2 不同方法的变体词还原效果

Pre@k	pre@1	pre@5	pre@10	pre@20
Huang et al. (2013)	37.09	59.40	65.95	70.22
Zhang et al. (2015)	38.17	66.38	73.07	78.06
Sha et. al. (2017)	36.50	62.50	75.90	84.70
AE-ECI	**41.88**	**72.07**	**82.33**	**88.89**

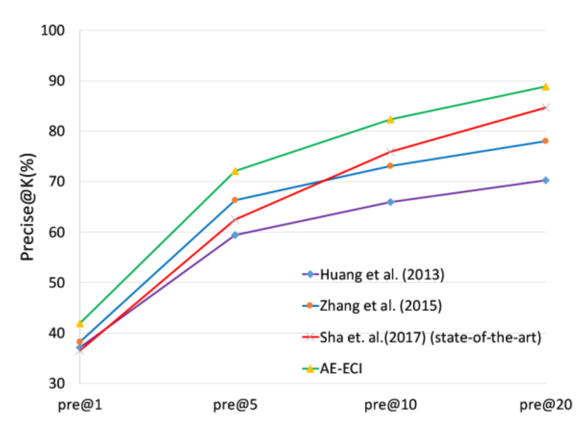


图 4 不同方法的变体词还原效果

3.4 参数讨论

a)λ。λ 是调节上下文在编码中影响的权值。本文取不同的 λ 进行实验，测试不同数值下变体词还原结果的 pre@1 值。由于首位还原结果若正确则意味着系统完全准确的还原了该变体词，因此 pre@1 指标最为重要，也最为代表性。因此此处本文仅测试最 pre@1 指标。其中 $\lambda=0$ 表示不添加有效上下文信息。如表 3 所示，本文可以发现添加有效上下文信息对于变体词还原任务是有提升的，但 λ 过大也会影响效果。

表 3 λ 对变体词还原效果的影响

λ	0.0	0.1	0.5	1.0
pre@1	39.88	41.31	**41.88**	40.31

b)**自编码器深度和编码维数。**自编码器隐层的维数 d 即编码表示的维数，深度即自编码器隐层的层数。本文取不同的深度和维度组合进行实验验证，结果如表 4 所示。可以看出维度过小或过大都会影响效果。可能的原因是维数过小不足以抽象输入的特征，维数过大导致神经网络参数过多，难以训练到好的效果。深度的取值也类似，可以发现深度较小时对效果影响不大，但网络过深也同样影响最终效果。

表 4 自编码器深度和维数对变体词还原效果的影响

维数	50	100	200	100	100
深度	3	3	3	2	5
pre@1	39.60	**41.88**	41.59	41.31	41.45

c)**窗口大小和有效上下文词项数量。**在抽取有效上下文信息时，选择不同的窗口大小 *wd* 和上下文词项数量 K 的组合来观察效果，结果如表 5 所示。可以看出，在正常的取值范围内，窗口大小和词项数量对变体词还原的结果影响不大。

表 5 窗口大小和有效上下文词项数量对变体词还原效果的影响

wd	5	10	20	50
K	10	20	10	20
pre@1	40.31	**41.88**	**41.88**	41.59

4 结束语

本文利用词项间点互信息来筛选有效上下文信息，并使用联合上下文的自编码器模型来融合词项及其有效上下文信息，生成联合编码，来完成变体词还原任务。自编码器是无监督模型，可以很好地对大量未标注数据进行训练，减少了人工标注的过程。根据点互信息筛选出的有效上下文信息可以更准确的发现变体词与目标词之间共有的、有特点的上下文，从而提高了变体词还原的准确性。

参考文献：

[1] Wong K F, Xia Y. Normalization of Chinese chat language [J]. Language Resources and Evaluation, 2008, 42 (2): 219-242.

[2] Xia Y, Wong K F, Li W. A phonetic-based approach to Chinese chat text normalization [C]// Proc of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. 【S. l. 】: Association for Computational Linguistics, 2006: 993-1000.

[3] Sood S O, Antin J, Churchill E F. Using crowdsourcing to improve profanity detection [C]// Proc of AAAI Spring Symposium: Wisdom of the Crowd. 2012: 06.

[4] Wang A, Kan M Y. Mining informal language from chinese microtext: joint word recognition and segmentation [C]// Proc of the 41th Annual Meeting of the Association for Computational Linguistics. 2013: 731-741.

[5] Han B, Cook P, Baldwin T. Automatically constructing a normalisation dictionary for microblogs [C]// Proc of Joint Conference on Empirical Methods in Natural Language Processing And Computational Natural Language Learning. Association for Computational Linguistics, 2012: 421-432.

[6] Li Z, Yarowsky D. Mining and modeling relations between formal and informal Chinese phrases from web corpora [C]// Proc of Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008: 1031-1040.

[7] Sha Y, Shi Z, Li R, *et al.* Resolving entity morphs based on character-word embedding [J]. Procedia Computer Science, 2017, 108: 48-57.

[8] Huang H, Wen Z, Yu D, *et al.* Resolving Entity Morphs in Censored Data [C]// Proc of the 41th Annual Meeting of the Association for Computational Linguistics. 2013: 1083-1093.

[9] Zhang B, Huang H, Pan X, *et al.* Context-aware Entity Morph Decoding [C]// Proc of the 43th Annual Meeting of the Association for Computational Linguistics. 2015: 586-595.

[10] Hiruncharoenvate C, Lin Z, Gilbert E. Algorithmically bypassing censorship on sina weibo with nondeterministic homophone substitutions [C]// Proc of the 9th International AAAI Conference on Web and Social Media. 2015: 150-158.

[11] Zhang B, Huang H, Pan X, *et al.* Be Appropriate and Funny: Automatic Entity Morph Encoding [C]// Proc of the 42th Annual Meeting of the Association for Computational Linguistics. 2014: 706-711.

[12] Sha Y, Shi Z, Li R, *et al.* Resolving Entity Morphs based on Character-Word Embedding [J]. Procedia Computer Science, 2017, 108: 48-57.

[13] Amiri H, Resnik P, Boyd-Graber J, *et al.* Learning text pair similarity with context-sensitive autoencoders [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics. 2016: 1882-1892.

[14] Zhou L, Zhang D. NLPIR: A theoretical framework for applying natural language processing to information retrieval [J]. Journal of the Association for Information Science and Technology, 2003, 54 (2): 115-123.

[15] Manning C D, Surdeanu M, Bauer J, *et al.* The Stanford corenlp natural language processing toolkit [EB/OL]. https://nlp. stanford. edu/pubs/StanfordCoreNlp2014. pdf. 2014: 55-60.

[16] Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[17] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.